



DevSecOps Software Factory Cost Estimation: Squaring the Circle

Cheryl Jones

cheryl.l.jones128.civ@army.mil

Joint IT and Software Cost Forum 2021
September 14 – 17, 2021





The Cost Estimator's Dilemma

Imagine:

- Cost estimating the first mass produced auto
- Cost estimating the first P-80 fighter aircraft production
- Cost estimating the first computerized IRS tax project



Agenda

- What makes a good cost estimate?
- What are traditional software cost estimation approaches and what do they depend on?
- How is DevSecOps cost estimation different from traditional software development estimation?
- What are the cost implications of a Software Factory?
- What is an approach to thinking about costing DevSecOps?



What Makes a Robust Software Cost Estimate?

- Ingredients of a good cost estimate - good CE process, reliable and validated CE data, low uncertainty in development process (follows standard practices)

Characteristics of a Reliable Cost Estimate

Comprehensive

The cost estimates should include costs of the program over its full life cycle, provide a level of detail appropriate to ensure that cost elements are neither omitted nor double counted, and document all cost-influencing ground rules and assumptions

Well-documented

The cost estimates should be supported by detailed documentation that describes the purpose of the estimate, the program background and system description, the scope of the estimate, the ground rules and assumptions, all data sources, estimating methodology and rationale, and the results of the risk analysis. Moreover, this information should be captured in such a way that the data used to derive the estimate can be traced back to, and verified against their sources.

Accurate

The cost estimates should be based on an assessment of most likely costs, and adjusted properly for inflation. Estimates should be grounded in documented assumptions and a historical record of cost estimating and actual experiences on other comparable programs. In addition, the estimates should be updated to reflect any changes.

Credible

The cost estimates should discuss any limitations of the analysis because of uncertainty, or biases surrounding data or assumptions. Risk and uncertainty analysis should be performed to determine the level of risk associated with the estimate. Further, the estimate's results should be crosschecked against an independent cost estimate to determine whether other estimating results produce similar results.

Underlying Assumptions & Constraints

- Reliable Development Process & Data exist
- Underlying operational context is well understood
- Uncertainty is generally bounded



Useful Software Cost Estimating Approaches

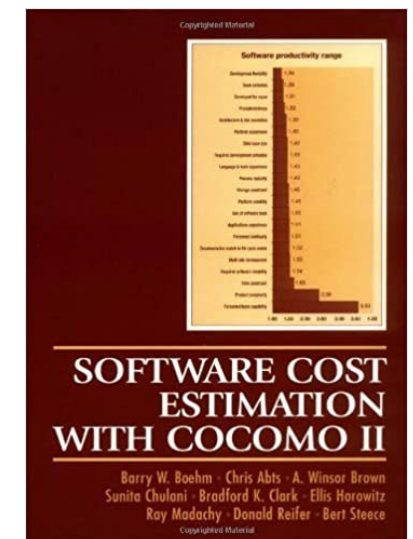
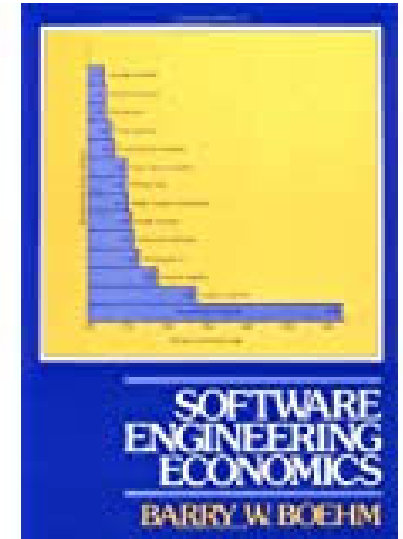
- Historical cost estimation approaches
 - How did they come about and what do they all have in common?
 - Parametric modeling
 - Activity-based modeling
 - Analogy-based estimates
 - Simple estimating relationships
 - It took years of experience to get good estimates, and they work better in some contexts than others
 - Based on domains, different approaches address different development approaches
 - Selection of different measures for cost estimation
 - Based on experience of what drives cost
 - Dependent on data

- What do they have in common?



Example: COCOMO® Model Evolution

- First COCOMO Software Cost Estimation model published in 1981
 - Data collection on 61 systems
 - Limitations due to small dataset
- Model evolved to address new development paradigms
 - COCOMO II (updated COCOMO 81)
 - COQUALMO (software quality model)
 - COINCOMO (incremental estimation)
 - COCOTS (COTS-based product development)
 - COSYSMO (Systems Engineering estimation)
 - Size measures have evolved over time, e.g. from estimated Lines of Code counts to allow other size measures, e.g. function points
- *The original COCOMO evolved into a family of estimation approaches, the same has to happen for DevSecOps estimation approaches*



Example: COCOMO[®] Estimation Scope

- The model is comprehensive
 - It estimates cost for a specific set of **life cycle phases** and **phase activities**
 - The model was calibrated to **technical and personnel data** from the software development (size, complexity, experience, etc.)
- The model is **well documented and fully open** so its estimates can be fully understood and explained
- The model is accurate for its purpose
 - It has been **updated over time** and continues to evolve
 - Accepts any labor rate
- Credible
 - Supports sensitivity analysis
 - Provides estimation uncertainty
- *DevSecOps cost estimation needs the same characteristics*



What Happens If Cost Estimating Context Radically Changes?

- Uncertainty increases across the board
 - in estimating process
 - in data reliability
 - in development process
- What do we know? What do we not know?
- For example, analogies break down
 - How many large scale defense software developments have been done with DevSecOps?
 - How many have published their cost estimates and supporting data?



DevSecOps is a Major Context Change

- Integrate development, system integration, and operation teams to **improve the collaboration process** and overall efficiency
- Manage application, workflow, and system integration design, development, deployment, modernization, and retiring/sunsetting
- Operate and maintain the on-premises and cloud data center environments, server operating systems, database administration, and platform support
 - Large (shared) infrastructure dependencies! Who owns it? Who pays for it?
- Support the end-to-end application and system integration lifecycle to include hardware and software: plan, develop, build, test, release, deliver, deploy, operate, and monitor, repeat
 - Automate all repeated processes, especially testing
- Embed cybersecurity into every stage of the system development process from the beginning of the effort



Poll Question

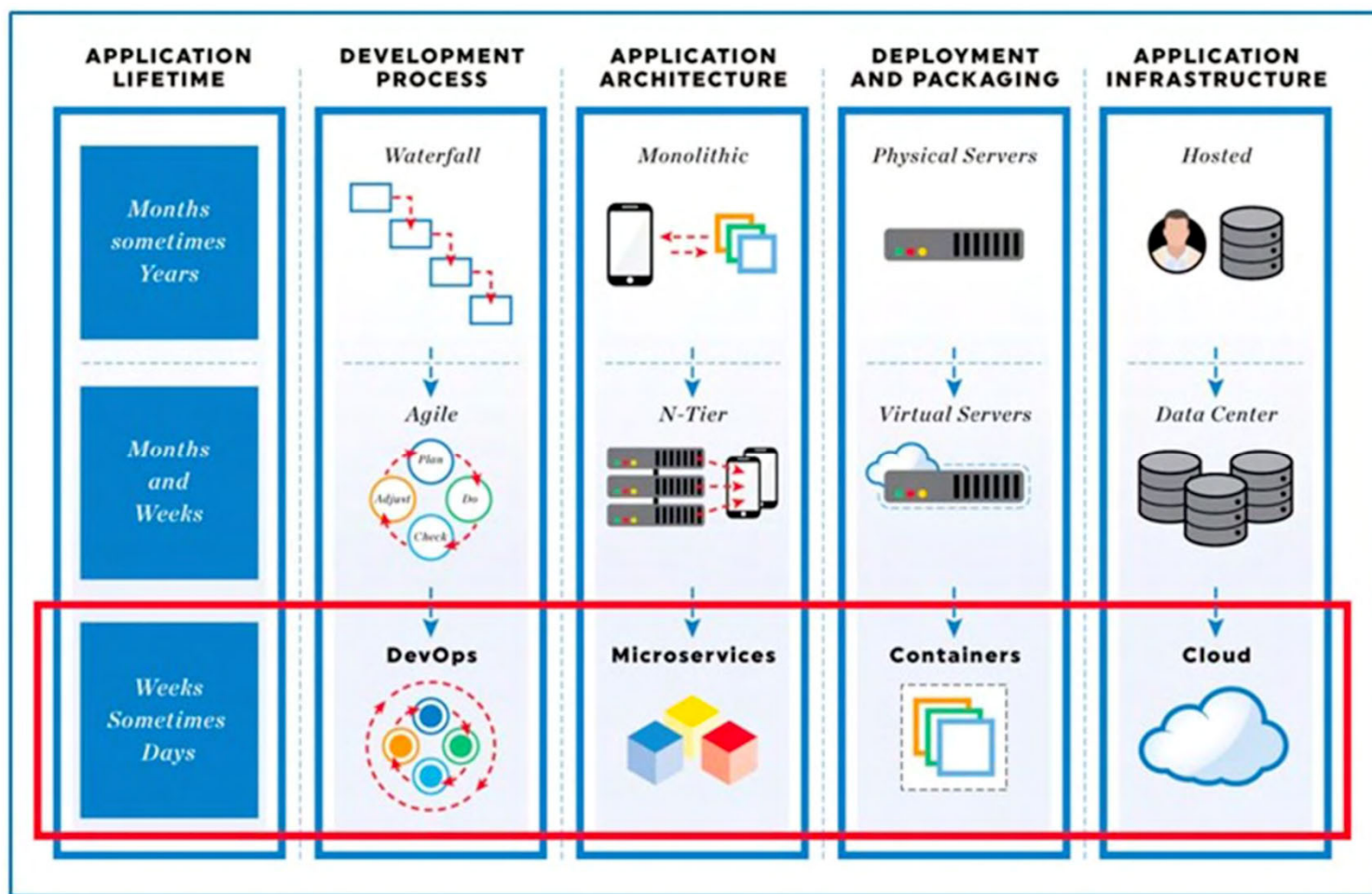
- Are you estimating DevSecOps?
 1. Yes
 2. No

- Are you using historical data as a basis of estimate?
 1. Yes
 2. No

- Are you receiving historical data from the software supplier?
 1. Yes
 2. No



Differences From Current DoD SW Development



Source: Chaillan, Nicolas, "DoD Enterprise DevSecOps Initiative", Cyber Security & Information Analysis Center (CSIAC) presentation, Aug. 8, 2019



DevSecOps Assumptions

1. A **certified and monitored cloud environment** exists
 - a. A cloud hostable anywhere Software Factory exists
 - b. A Software Factory (both Open Container Initiative (OCI) & Cloud Native Computer Foundation (CNCF) compliant and certified) exists
2. The DevSecOps architecture is **infinitely scalable**
 - a. Open sources vs COTS risks are assessed, mitigated or assumed
 - b. DevSecOps pipelines and patterns are defined
3. There are **no major cultural issues** related to DevSecOps
 - a. End-users have bought-into DevSecOps approach and are available throughout the project
 - b. Stakeholders have bought-into DevSecOps approach

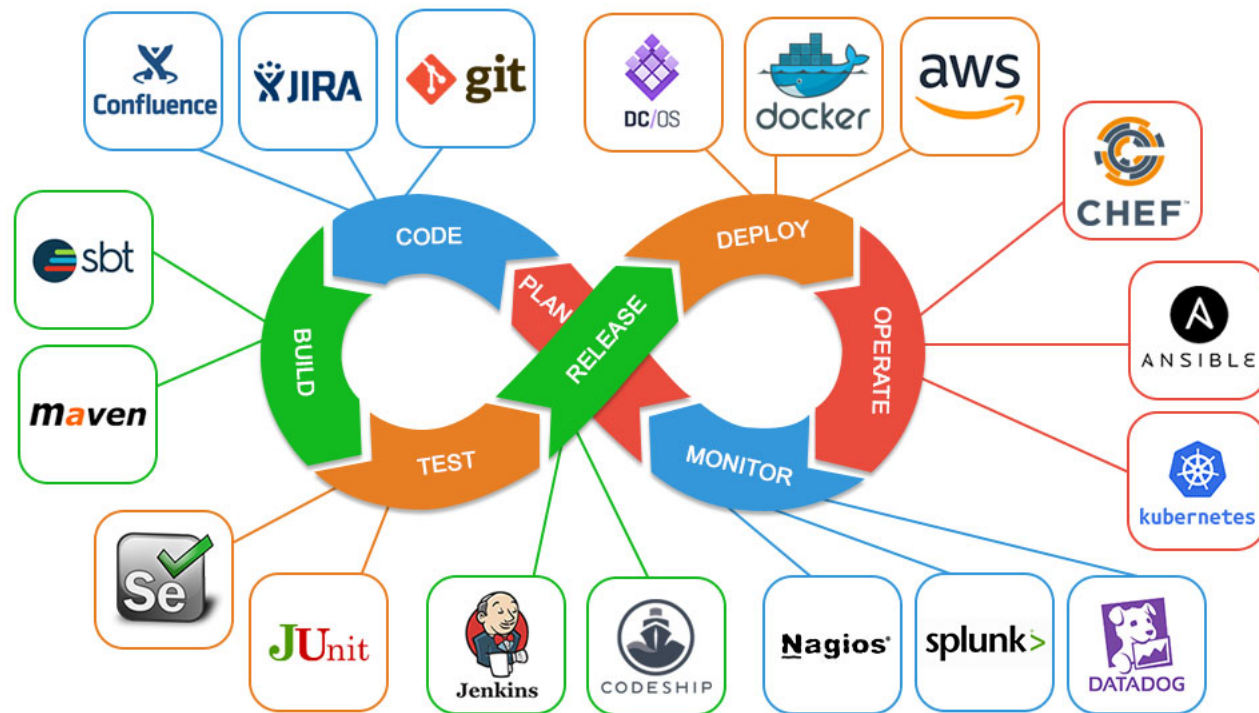


Estimation Challenge

- DevSecOps is based on the principle of Continuous Integration (CI) and Continuous Delivery (CD)
- Multiple development pipelines create, enhance, and maintain different software products independently and concurrently using a **Software Factory construct**
- This construct relies on **task automation for repeatable tasks** thus reducing workload (and cost), improving quality, and increasing the speed of delivery
- Automation is enabled with software tools
 - The cost of startup and buildout of a software factory is driven by effort and supporting tool costs
 - The cost will vary with the degree of implementation
- A Software Factory has an initial start-up and an ongoing cost



Task Automation for Repeatable Tasks



Start-up and Infrastructure Costs!



The Cost Estimator's New Dilemma

- Finding of the Software Sustainment and Maintenance of Weapons Systems for the United States Air Force Workshop held in March 2020:

*“DoD’s push for **agile software development practices** will mean **that cost benchmarks will no longer be hard and fast**, and programs will not know the full cost of development and sustainment upfront due to agile practices’ need for flexibility.”*

- What does that mean for cost estimators?



Poll Question

- Are you estimating DevSecOps cost? (pick all that apply)
 1. The software factory
 2. Minimum Viable Product
 3. Continuous pipeline



Estimation Paradigm -1

Selection

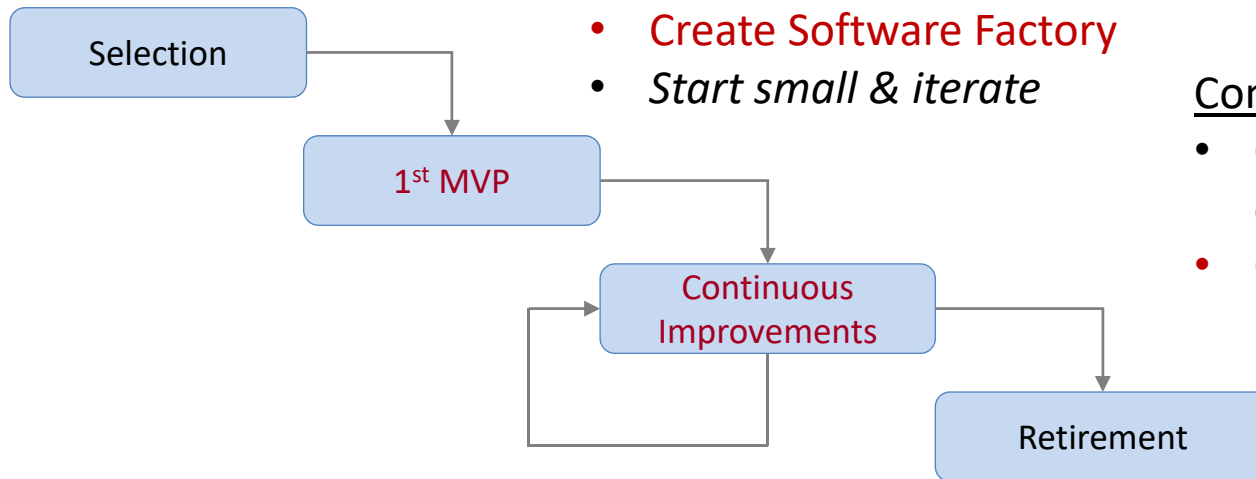
- Program selection
- Migration to Pipeline strategy
- Infrastructure strategy
- Training plans

1st Minimum Viable Product (MVP)

- **Create Software Factory**
- *Start small & iterate*

Continuous Improvements

- Continuous software enhancement & repair
- **Continuous Factory/ pipeline buildouts**



Retirement

- Pipeline shutdowns

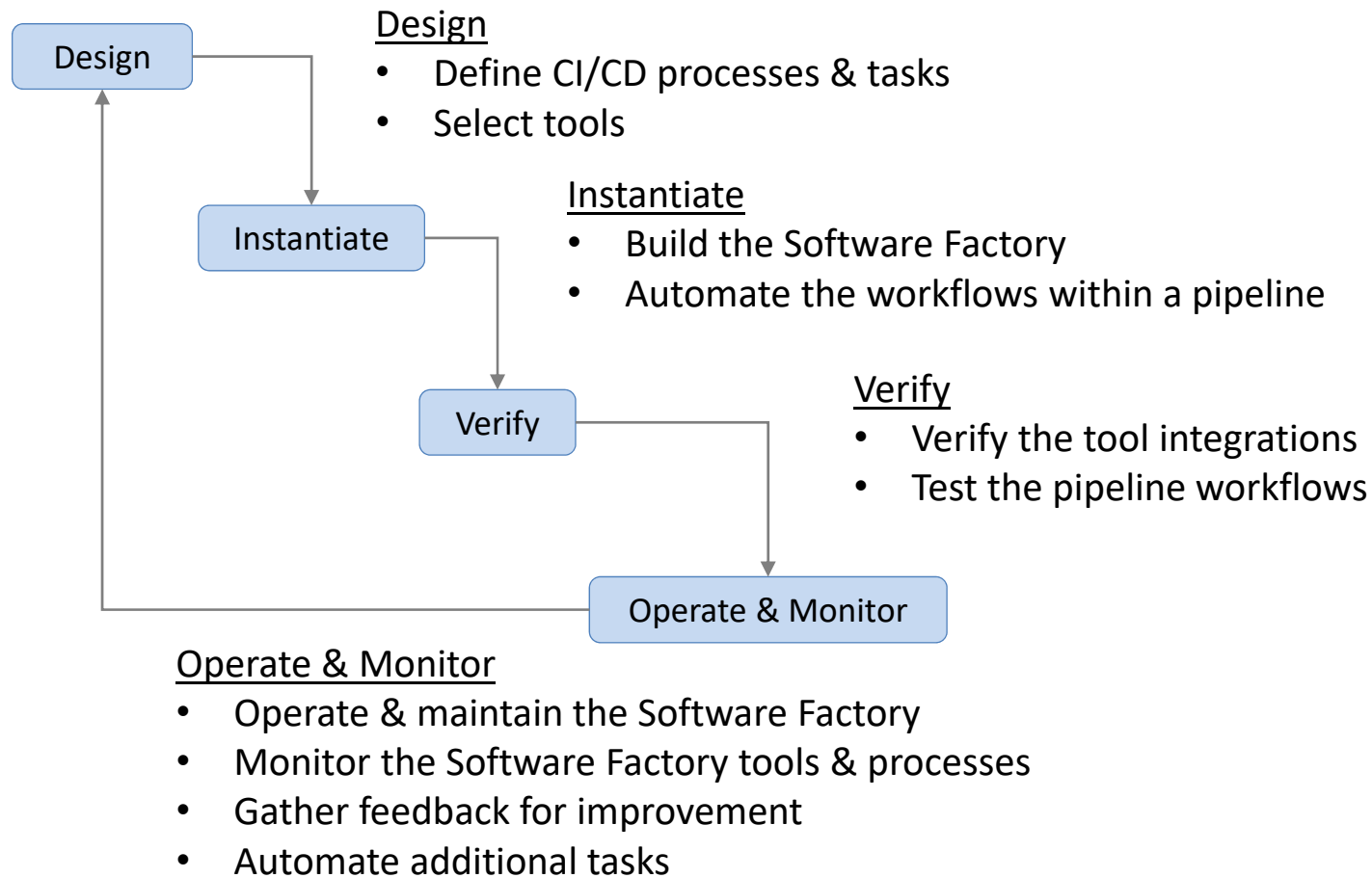


Estimation Assumption #1

- Cost estimation focus is
 - Delivery of the 1st MVP, and/or
 - Continuous Improvements
- Historically, improving cost estimates are challenging
- Added cost complexity is due to the iterative buildout of each pipeline in the Software Factory
 - Normal software development costs change as automation is added
 - Process automation with software tool installation requires supporting infrastructure, another cost, e.g., cloud services, 3rd party software licenses, open source tools



Software Factory Phases



How does this impact a cost estimate's scope?

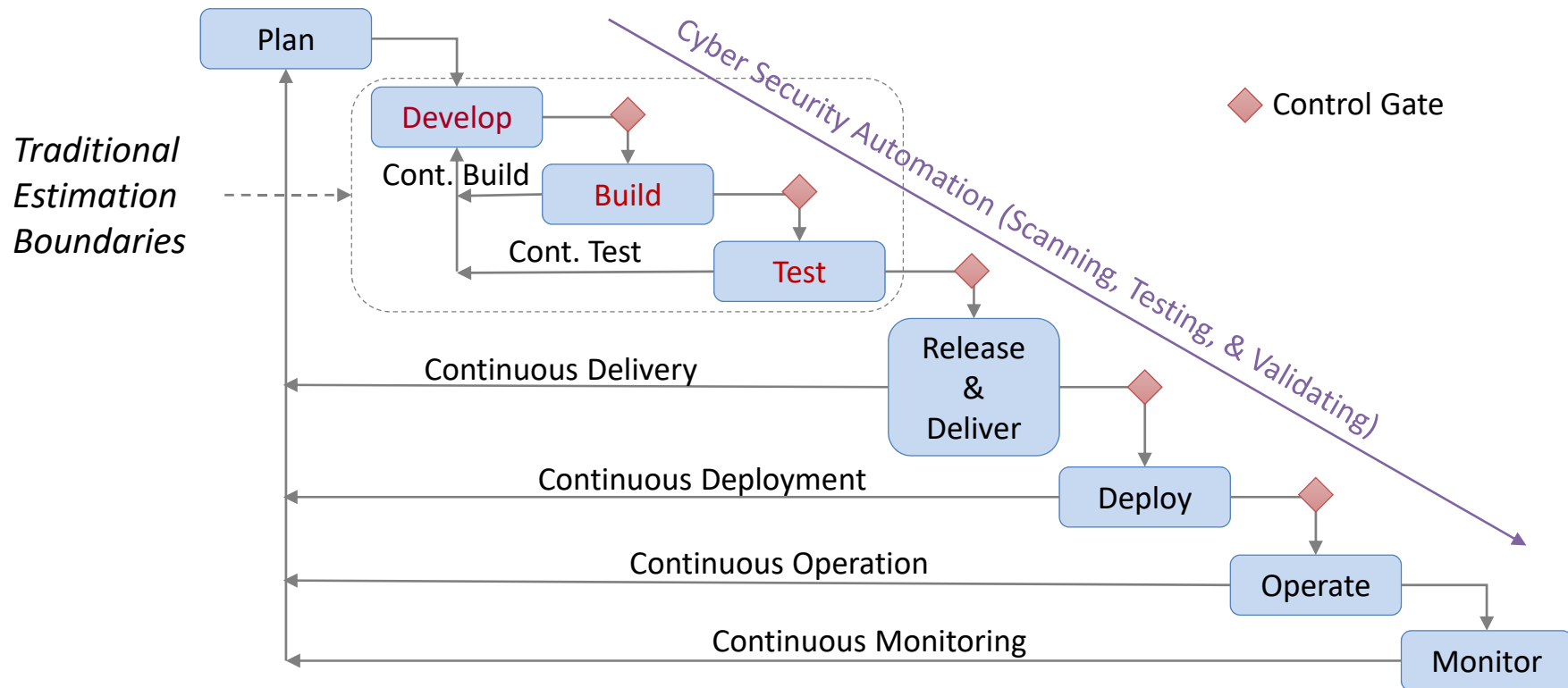


Estimation Assumption #2

- 1st MVP requires automating a workflow within a pipeline, e.g., Release
- More workflows in a pipeline that can be automated are added successively
- A Software Factory consists of multiple pipelines (next slide)
- Automation instantiation costs & learning curve disruption will continue until a pipeline is fully implemented



DevSecOps Pipeline Phases



- To adopt a DevSecOps process successfully, implement it in multiple, iterative phases
- Start small with some tasks that are easy to automate, then gradually build up the capability and adjust the processes to match
- **Because both pipeline automation & software development are dynamic, it is challenging to predict cost**

Source: "DoD Enterprise DevSecOps Playbook", Department of Defense Chief Information Officer, Feb 5, 2021



Wrap-Up

- What is the cost estimator being asked to estimate in DevSecOps?
 - Factory costs?
 - MVP costs?
 - Pipeline costs? Which pipeline stages?
 - All of the above?



Squaring the Circle

- What we are trying to do to help DevSecOps cost developers?
- Army Software Maintenance Initiative has been identifying the risks, assumptions, and constraints of DevSecOps
- Developing an approach to help cost estimators understand the elements that need be considered for DevSecOps

We have a once in a lifetime opportunity to materially improve DoD cost estimation



Contributors

Cheryl Jones

Measurement Analyst
US Army AFC
973-724-2644

James Judy

NISEC Division Chief
ODASA-CE
703-697-1612

Paul Janusz

Measurement Analyst
US Army AFC
973-724-4849

Dr. Robert Charette

President
ITABHI Corporation
(540) 972-8150

Dr. Bradford Clark

Vice President
Software Metrics, Inc.
(703) 754-0115

